Amendments to the Specification:

Paragraph beginning on page 1, line 16

Equalization is a well-known signal processing technique used to combat the Intersymbol Interference distortion imparted to the transmitted signal by the channel whereby the receiver attempts to compensate for the effects of the channel on the transmitted symbols. An equalizer attempts to determine the transmitted data from the received distorted symbols using an estimate of the channel that caused the distortions. Examples of commonly used types of equalizers include the maximum likelihood sequence estimation (MLSE) equalizer that utilizes the well known Viterbi Algorithm (VA), linear equalizer and decision feedback equalizer (DFE). In communications systems where ISI arises due to partial response modulation or a frequency selective channel, a maximum likelihood sequence estimation (MLSE) equalizer is optimal.

Paragraph beginning on page 1, line 24

Many modern equalization techniques, however, utilize very complex signal processing algorithms to achieve acceptable levels of performance. Practically, most of these techniques can only be performed on expensive high powered digital signal processors. With the ever pressing demand to make communications enabled products smaller, cheaper and high performing, it would be desirable to implement the entire communications hardware in silicon, such as in an Application Specific ~~Integrate~~ Integrated Circuit (ASIC) or the like. To place the complex, sophisticated equalization techniques of the prior art on an ASIC, however, would be nearly impossible in terms of gate count and cost, given today's available gate densities and chip sizes. Prior art equalization techniques require extremely large processing resources and memory in order to implement them in silicon.

Paragraph beginning on page 3, line 6

Many types of channels, especially the power line media, typically suffer from Intersymbol Interference due to the pulse spreading effects of the channel. The iterative equalization technique of the present invention provides a solution to the communication problems imposed by the characteristics of the power line media, e.g., Intersymbol Interference, etc. In addition, the technique is well suited for implementation in integrated circuit form, e.g., ASIC, etc. thus enabling [[the]] communications over noisy channels in receivers employing a wide range of communication technologies.

Paragraph beginning on page 8, line 10

Many types of channels, especially the power line media, typically suffer from Intersymbol Interference due to the pulse spreading effects of the channel. The iterative equalization technique of the present invention provides a solution to the communication problems imposed by the characteristics of the power line media, e.g., Intersymbol Interference, etc. In addition, the technique is well suited for implementation in integrated circuit form, e.g., ASIC, etc. thus enabling [[the]] communications over noisy channels in receivers employing a wide range of communication technologies.

Paragraph beginning on page 10, line 28

Equalization is a well known technique used to combat intersymbol interference whereby the receiver attempts to compensate for the effects of the channel on the transmitted symbols. An equalizer attempts to determine the transmitted data from the received distorted symbols using an estimate of the channel that caused the distortions. Stated in other words, equalization is a technique [[use]] <u>used</u> for removing deterioration caused by the channel from received symbols, wherein the deterioration is usually caused by intersymbol interference. The ISI may be caused by reflections, multipath, fading and the effects of a non-flat transfer function for the channel.

Paragraph beginning on page 11, line 20

[[h_n]] $\underline{h_i}$ represents the impulse response of the channel;

Paragraph beginning on page 12, line 8

[[B_n]] $\underline{B_{j-i}}$ represents the bits transmitted;

Paragraph beginning on page 12, line 9

[[h_n]] $\underline{h_i}$ represents the impulse response of the channel;

Paragraph beginning on page 12, line 12

Note that the entity y_j represents the impulse response of the symbol n (including additive noise) only whereby the impulse response of the previous symbols has been subtracted out. This operation 'cleans' the intersymbol interference from previous symbols but not from future symbols, so still a large amount of ISI is left. Note that, for purposes of calculating Equation 2, the soft decisions (as are computed below) are used for the previous symbols while zero is used for the case j-i<0<u>.</u>

Paragraph beginning on page 12, line 18

Once the modified received symbol y_j is calculated, the estimate of the current symbol (bit) $\widetilde{B}_n$ can then be calculated using the following expression

$$\widetilde{B}_n = f(\sum_{i=0}^{m} y_{i+n} \cdot h_i^*) \qquad (3)$$

where $f(\ )$ is a soft decision function (e.g., hyperbolic tangent). The slope of the decision function preferably increases with the iterations. Thus, the modified received symbol y_n is processed through a matched filter operation (i.e. a correlation function) which functions to estimate the symbol. Note that in the case of QPSK the soft decision is applied separately on the real and imaginary ~~component~~ components. In the case of QAM or other modulation, the soft decision needs to be modified accordingly.

Paragraph beginning on page 13, line 16

[[x_n]] x_{i+n} represents the input to the equalizer;

Paragraph beginning on page 13, line 17

[[h_n]] h_i represents the impulse response of the channel;

Paragraph beginning on page 14, line 19

In the next iteration, Equation 2 is modified to yield

$$y_j = x_j - \sum_{i=j-n+1}^{m} \widetilde{B}_{j-i} \cdot h_i - \sum_{i=0}^{j-n-S} \widetilde{B}_{j-i} \cdot h_i \text{, j=n...n+m-1} \qquad (7)$$

where the third term in the equation represents the contribution of future symbols to x_j j and S represents the number of neighbors to be ~~processing~~ processed using linear equalization. These S symbols ~~are making~~ make up a sliding window.

Paragraph beginning on page 18, line 5

The influence of all the symbols already estimated apart from a window of S-1 neighbors, is subtracted from the current symbol using the channel estimate and the m-1 previous and m-1 future soft decisions (step 34). The soft decision is ~~done~~ calculated using Equations 7 and 8 in step 35. The error difference is then calculated between the symbols estimated during this iteration and those of the previous iteration (step 36). The error difference is compared to a predetermined threshold (step 38), and if greater, another iteration is performed. The soft decisions for the symbols are updated (step 40) and the method repeats steps 34, 36 and 38. The method terminates when the error difference is less than the predetermined threshold. Note that alternatively, the number of iterations may be fixed as in the example implementation presented hereinbelow.

Paragraph beginning on page 18, line 29

The iterative equalizer, generally referenced 18, comprises a modified received symbol calculator 40, matched filter 42, soft decision calculator 44, quantizer 46, multiplexer 48, summer 50 and delay 52. The equalizer is operative to generate soft symbol decisions $\widetilde{B}_n$ using as input the received symbol x_n samples, channel impulse response h_i from the channel estimator and h_{10} computed ~~from it~~ therefrom.

Paragraph beginning on page 19, line 20

Note that the length of the equalizer is in this example is equal to 64 samples. The ratio between the sample rate and the symbol rate (or bit rate) [[is]] in this example is four samples per symbol while the number of iterations is also fixed at [[4]] four. The equalizer structure disclosed herein can be modified by one skilled in the art to accommodate different numbers of iterations and different number of samples per symbol, both of them not necessarily being equal to each other.

Paragraph beginning on page 20, line 24

Each row comprises 16 multipliers and 16 adders. The top row uses coefficients h_{63}, h_{59}...h_3; while the next row uses coefficients h_{62}, h_{58}...$h2$; the following row coefficients h_{61}, h_{57}...h_1; and the bottom row coefficients h_{60}, h_{56}...h_0. Thus, at each sample time, the influence of the previously processed symbols is subtracted out. Sixteen subtractions are performed for each row. Each sample, however, is shifted through its row four times resulting in four symbol subtractions per input symbol. Each such subtraction is timed such that it is carried out on the sample belonging to the next iteration. Likewise, the CORR calculation is timed such that it is available at the ~~right~~ proper time in the ~~right~~ proper iteration. In this fashion, the sample shift register array implements the iterative subtraction described supra.

Paragraph beginning on page 20, line 33

It is important to point out that the position in each row indicated by reference numeral 73 does not have a multiplier/adder associated with it. Once the correction impulse response of the previously processed symbol has been iteratively subtracted, the resulting modified received symbol 41 is effectively passed through a matched filter via multipliers 84. A total of 64 taps of the sample shift register array contents are connected to multipliers 84 which are operative to generate the product of the channel response and the modified received symbol. A tap is connected to every fourth register in each row. Sixteen multipliers are associated with each shift register row, wherein coefficients h_{63}, h_{59}...h_3 are associated with the first row; coefficients h_{62}, h_{58}...$h2$ are associated with the second row; coefficients h_{61}, h_{57}...h_1 are associated with the third row; and coefficients h_{60},

$h_{56} \ldots h_0$ are associated with the fourth (bottom) row. The apparatus 80 thus effectively implements Equation 4 on the current ~~content~~ contents of the modified received signal.

Paragraph beginning on page 21, line 11

The outputs of the multipliers may be stored in product registers (not shown). Summer 88 is adapted to add all 64 product results so as to yield a matched filter soft output 43 for the current symbol, denoted MF(n).

Paragraph beginning on page 21, line 25

[[and]]

Paragraph beginning on page 22, line 21

The result $\widetilde{B}_{n-1}$ is then (optionally) input to the soft decision function 45. Note that some optimization in the number of calculations can be obtained by substituting ~~Equation~~ Equations 12 and 13 into Equation 14.